

A method of configuring computer peripherals.

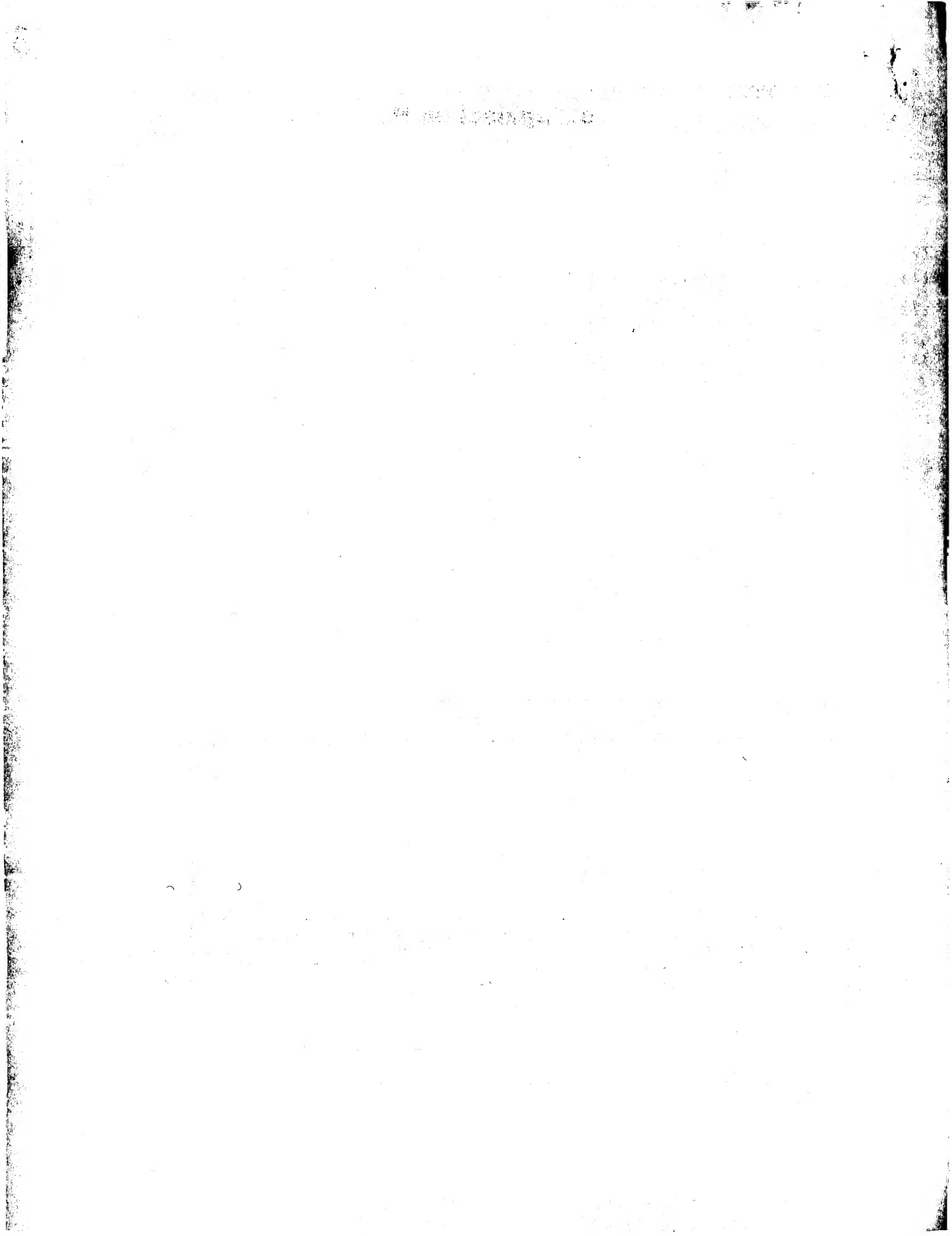
Patent Number: EP0663638
Publication date: 1995-07-19
Inventor(s): AMSDEN JONATHAN D (US); KAISNER JEFFERY WAYNE (US); TERRELL MICHAEL R (US); BURKE THOMAS C (US); TODD DAVID K (US)
Applicant(s): AT & T GLOBAL INF SOLUTION (US)
Requested Patent: ☐ EP0663638, B1
Application Number: EP19940309649 19941221
Priority Number (s): US19930176022 19931230
IPC Classification: G06F13/38; G06F15/16
EC Classification: G06F13/38A2, G06F15/177
Equivalents: DE69428512D, DE69428512T, ☐ JP7210488
Cited Documents: US5237689; EP0192066; EP0048781

Abstract

A method for the configuration of peripheral adapters (14,16,18) in computer systems. The method examines a flag in a Configuration Space of the peripheral adapter (14,16,18) to determine if user-selectable configuration options are required for the peripheral adapter (14,16,18). If the flag is set, then a configuration file (34) is retrieved and interpreted to determine how the peripheral adapter should be configured and to determine how the configuration options should be presented to the user. Such configurations options may be conditional and based on previous user selections. Once the user's selections have been made, the method updates registers (36) in the peripheral adapter (14,16,18)

with the selected option values. 

Data supplied from the esp@cenet database - I2





① BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

② Übersetzung der
europäischen Patentschrift

⑦ EP 0 663 638 B 1

⑩ DE 694 28 512 T 2

⑥ Int. Cl. 7:
G 06 F 13/38
G 06 F 15/16

② Deutsches Aktenzeichen: 694 28 512.9
⑤ Europäisches Aktenzeichen: 94 309 649.5
⑥ Europäischer Anmeldetag: 21. 12. 1994
⑦ Erstveröffentlichung durch das EPA: 19. 7. 1995
⑦ Veröffentlichungstag
der Patenterteilung beim EPA: 4. 10. 2001
④ Veröffentlichungstag im Patentblatt: 29. 5. 2002

③ Unionspriorität:
176022 30. 12. 1993 US
⑦ Patentinhaber:
NCR International, Inc., Dayton, Ohio, US
⑦ Vertreter:
v. Bezold & Sozien, 80799 München
⑧ Benannte Vertragsstaaten:
DE, FR, GB

⑦ Erfinder:
Terrell, Michael R., Easley, US; Kaisner, Jeffery
Wayne, Easley, US; Amsden, Jonathan D., Easley,
US; Burke, Thomas C., Liberty, US; Todd, David K.,
Greenville, US

⑤ Konfigurationsverfahren für Rechnerperipheriegeräte

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

DE 694 28 512 T 2

DE 694 28 512 T 2

BESCHREIBUNG

Die Erfindung bezieht sich generell auf Verfahren zur Konfiguration von Computersystemen und speziell auf ein Verfahren zur Konfigurierung eines peripheren Adaptors.

Das periphere Komponenten-Interface (PCI) Local Bus Specification Production Version, Reversion 2.0 vom 30.04.1993 definiert die Standardmechanismen, welche die Verbindung von peripheren Geräten an einem PCI-Lokalbus ermöglichen. Die PCI-Spezifikation befasst sich auch damit, wie periphere Geräte im System Ressourcenerfordernisse definieren und diese dem Computer mitteilen können. Es wird berücksichtigt, dass einige Geräte eine benutzerseitige Auswahl von Konfigurationsoptionen erfordern. Die PCI-Spezifikation behandelt jedoch nicht, wie ein peripheres Gerät gerätespezifische Konfigurationsoptionen an einem Systemdienst (utility) weiterleitet, so dass sie dem Benutzer zur Auswahl und dem Computer zur Konfigurierung des peripheren Gerätes dargeboten werden können.

Ohne einen Standardmechanismus für die Konfigurierung peripherer Geräte würde jeder Lieferant seinen eigenen speziellen Konfigurationsmechanismus mitliefern. Für PCI-Systeme, welche bereits einen Systemkonfigurationsdienst vorgesehen haben, beispielsweise wo das PCI-System einen MicroChannel™-(MCA) oder EISA-Bus enthält und die MCA- oder EISA-Konfigurationsdienste benutzt, wäre es nicht akzeptabel, von den Nutzern mehrere Konfigurationsdienste zu fordern, also jeweils einen für jedes der peripheren Geräte. Weiterhin wären die Lieferanten gezwungen, sich mit den Entwicklungs- und Managementproblemen herumschlagen, die mit den mehrfachen Konfigurationsdiensten für mehrfache Betriebsumgebungen verbunden sind.

Es ist eine Aufgabe der vorliegenden Erfindung, ein Verfahren zur Konfigurierung von Computer-Peripheriegeräten zu schaffen, welches die oben genannten Probleme überwindet.

Demgemäß sieht die Erfindung ein Verfahren zur Konfigurierung eines mit einem Computer verbundenen peripheren Adapters vor, mit den Schritten: (a) Feststellung, ob der periphere Adapter vom Nutzer wählbare Operationsparameter hat, (b) Zugreifen auf eine im peripheren Adapter gespeicherte Konfigurationsdatei, wenn der periphere Adapter vom Benutzer wählbare Operationsparameter hat, (c) Interpretieren des Inhalts der Konfigurationsdatei zur Identifizierung von Optionen für die vom Benutzer wählbaren Operationsparameter des peripheren Adapters, (d) Darstellen der Optionen der vom Benutzer wählbaren Operationsparameter des peripheren Adapters für einen Benutzer des Gerätes, (e) Akzeptieren einer Wahl einer der Optionen für jeden der vom Benutzer wählbaren Operationsparameter des peripheren Adapters seitens des Gerätebenutzers, und (f) Einstellen der Operationsparameter des peripheren Adapters entsprechend der gewählten Option.

Eine Ausführungsform der Erfindung sei nun im Rahmen eines Beispiels mit Bezug auf die beiliegenden Zeichnungen erläutert. Es zeigen:

- Figur 1 ein Blockdiagramm eines Konfigurationsbeispiels unter Verwendung des PCI-Busses;
- Figur 2 ein Blockdiagramm zur Veranschaulichung des Formates des Konfigurationsraumes (Configuration Space) in einem PCI-Adapter;
- Figur 3 ein Blockdiagramm zur Veranschaulichung des Formates des Erweiterungs-ROMs in einen PCI-Adapter;

Figur 4 Diagramme zur Veranschaulichung einer Konfigurations-Lese-Transaktion zwischen dem Prozessor und dem PCI-Adapter; und

Figuren 5A bis G gemeinsam ein Flussdiagramm zur ausgeführten Dienste-Programm ausgeführt werden.

Figur 1 stellt ein Blockdiagramm einer beispielhaften Konfiguration unter Verwendung eines Peripher-Komponenten-Verbindungs-(PCI)-Bus 10 dar. Der PCI-Bus 10 ist eine gegenständliche Verbindungsvorrichtung, welche für die Benutzung von peripheren Geräten oder Adaptern und Prozessor/Speicher-Systemen vorgesehen ist. Der PCI-Bus 10 ist als ein Standardinterface auf der Komponentenebene bestimmt, in genau derselben Weise, wie ISA-, EISA- oder MCA-Busse Standardinterface auf der Board-Ebene sind. Ebenso wie ISA-, EISA- und MCA-Busse ein gemeinsames I/O-Board-Interface zwischen verschiedenen Plattformen und verschiedenen Prozessorgenerationen bilden, ist der PCI-Bus 10 dazu gedacht, ein gemeinsames I/O-Komponenten-Interface über verschiedene Plattformen und verschiedene Prozessorgenerationen darzustellen. Eine vollständigere Diskussion des PCI-Busses 10 findet sich in dem Dokument PCI Local Bus Specification Production Version, Revision 2.0 vom 30. April 1993.

In Figur 1 ist ein Prozessor-/Speicher-Untersystem 12 an den PCI-Bus 10 angeschlossen, wie auch periphere Adapter als Audioboards 14, Videoboards 16, Videographikkontrollern 18 für Monitore 20, SCSI-Periphergeräte 22, LAN-Interfaces 24 und andere periphere Geräte 26. Eine Brückeninterface-Logik 28 koppelt den PCI-Bus 10 an einen Standarderweiterungsbus 30 und bildet den Zugang zwischen ihnen für verschiedene Erweiterungs-I/O-Boards 32. Für den Fachmann versteht es sich, dass diese Beispielskonfiguration keine Beschränkungen auf irgendeine spezielle Computerarchitektur bedeuten soll.

Wie nachstehend in weiteren Einzelheiten beschrieben wird, schlägt die Erfindung ein Konfigurationsverfahren für mit dem PCI-Bus 10 gekoppelte periphere Adapter vor. Die peripheren Adapter haben vom Benutzer wählbare gerätespezifische Konfigurationsoptionen, die durch eine PCI-Konfigurationsdatei (PCF) definiert sind, welche mit dem peripheren Adapter mitgeliefert wird. Zunächst greift der Prozessor auf einen von der PCF-Datei geforderten Merker in einem Konfigurationsraum (Configuration Space) in dem betreffenden peripheren Adapter zu. Wenn dieser Merker gesetzt ist, dann greift der Prozessor auf eine Erweiterungs-ROM im peripheren Adapter zu, um zu bestimmen, ob die PCF-Datei in dem Erweiterungs-ROM vorhanden ist. Wird sie dort nicht gefunden, dann prüft der Prozessor andere nicht-flüchtige Speicher, oder der Benutzer wird angewiesen, eine Diskette mit der PCF-Datei zu benutzen. Hat der Prozessor die PCF-Datei gefunden, dann werden die darin enthaltenen Passwörter interpretiert, um den Benutzer zur Auswahl unter verschiedenen Konfigurationsoptionen anzuweisen. Hat der Benutzer seine Wahl getroffen, dann aktualisiert der Prozessor die entsprechenden Register im Konfigurationsraum des peripheren Adapters mit den vom Benutzer gewählten Werten und speichert die Benutzerauswahl in einem nicht-flüchtigen Speicher für die nächste Umladung oder einen "Kaltstart" des Systems.

Das erfindungsgemäße Verfahren besitzt eine Vielzahl von Vorteilen gegenüber dem Stand der Technik. Ein Vorteil besteht in der Benutzung, Anforderungen oder Wünsche bezüglich der peripheren Geräte zu spezifizieren. Weiterhin ist die Erfindung unabhängig sowohl vom Betriebssystem wie auch von der Systemarchitektur. Die Formatierung der Konfigurationsdateien erfolgt unter Verwendung von ANSI-, ASCII-, EBCDIC- oder ISA 8859.1-Text, so dass sie nicht auf irgendeine spezielle Plattform beschränkt sind. Ferner ist die Syntax-Spezifikation der Konfigurationsdateien erweiterbar, so dass Revisionen leicht

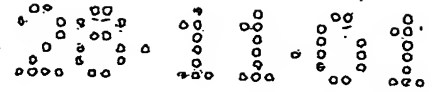
vorgenommen werden können, wobei rückwirkend die Kompatibilität mit früheren Versionen erhalten bleibt.

Ein anderer Vorteil besteht darin, dass die Syntax-Spezifikation eine herstellerkompatible (generic) Fähigkeit zur Übertragung gerätespezifischer Informationen aufweist. Die Syntax-Spezifikation ist einfach und besteht aus nur wenigen Passwörtern, so dass ein Lieferant herstellerspezifische Textfolgen zur Beschreibung der Konfigurationsregister, Funktionen, Merkmale und Optionen benutzen kann. Damit entfällt die Notwendigkeit einer langen und komplizierten Passwortliste für die Behandlung aller möglichen Konfigurationsoptionen.

Ein weiterer Vorteil besteht darin, dass die Erfindung nur eine minimale Auswirkung auf die PCI-Spezifikation hat, obwohl sie eine Erweiterung über diese hinaus umfasst. Die Erfindung benötigt nur ein einziges Bit im PCI-definierten Konfigurationsraum, um diejenigen Geräte zu identifizieren, welche die benutzerseitige Auswahl der Konfigurationsoptionen erlauben. Darüber hinaus beeinflusst die Verwendung dieses einzigen Bits die peripheren Adapter, welche nicht konfigurierbar sind, nicht nachteilig, weil das Bit normalerweise zu einem Wert verzögert wird, der bedeutet, dass keine Konfigurationsoptionen bestehen.

Noch ein anderer Vorteil der Erfindung besteht darin, dass sie eine automatische Konfiguration peripherer Adapter ohne Eingriff seitens des Benutzers erlauben. Beispielsweise kann die erste "Wahl" als Grundwahl programmiert werden. Wenn nur eine einzige "Wahl" für eine Konfigurationsoption verfügbar ist, wird außerdem erfindungsgemäß automatisch diese Wahl ohne Eingriff des Benutzers gewählt.

Gemäß einem weiteren Vorteil erlaubt es die Erfindung, dass die Konfigurationsdatei in dem Erweiterungs-ROM des peripheren

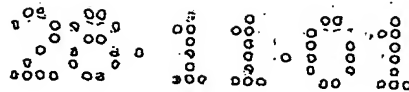


Adapters angesiedelt ist. Diese Eigenschaft erübrigt Disketten, welche die Konfigurationsdatei enthalten.

Indem sie eine standardisierte Methode zur Konfigurierung peripherer Adapter bietet, erübrigt die Erfindung die Notwendigkeit, dass der Lieferant seine eigenen Konfigurationsdienste erstellt und einbaut. Fernerhin erübrigt es sich, dass der Benutzer mit mehreren Konfigurationsdiensten für mehrere periphere Adapter, die in einem Computersystem vorhanden sein können, arbeiten muss.

Figur 2 zeigt ein Blockdiagramm, welches das Format des Konfigurationsraumes 34 in dem peripheren Adapter veranschaulicht. Der Konfigurationsraum 34 weist einen 256 Byte-Speicher auf, der in einen vorbestimmten Kopfbereich 36 und einen geräteabhängigen Bereich 38 unterteilt ist. Der Kopfbereich 36 beginnt bei der Adresse 00_{16} , erstreckt sich über eine Länge von 64 Bytes und definiert eine Mehrzahl von festen Registern oder Feldern. Der geräteabhängige Bereich 28 beginnt bei der Adresse 40_{16} , erstreckt sich bis zur Adresse FF_{16} und unterstützt gerätespezifische Register und Funktionen. Für weitere Informationen über das Format des Konfigurationsraumes 34 sei auf das Dokument PCI Local Bus Specification Production Version, Reversion 2.0 vom 30. April 1993 verwiesen, auf welches hiermit Bezug genommen sei.

Bei der Erfindung enthält eines der reservierten Register (bei 28_{16} , $2C_{16}$, 34_{16} oder 38_{16}) im Kopfbereich 36 das für die PCF-Datei erforderliche Feld, vorzugsweise ein Feld für ein einziges Bit. Ein Wert von "1" in dem für die PCF-Datei benötigten Feld zeigt an, dass eine Benutzerauswahl von einer oder mehreren Konfigurationsoptionen möglich ist. Ein Wert von "0" in dem von der PCF-Datei benötigten Feld zeigt an, dass der periphere Adapter keine Benutzerauswahl irgendeiner Konfigurationsoption benötigt.



Typischerweise beziehen sich ein oder mehrere Register des geräteabhängigen Bereiches 38 des Konfigurationsraumes 34 auf vom Benutzer wählbare Konfigurationsoptionen. In Abhängigkeit von den in den Registern gespeicherten Werten kann der periphere Adapter auf verschiedene Weisen arbeiten.

Figur 3 stellt ein Blockdiagramm dar, welches das Format des Erweiterungs-ROMs 40 im peripheren Adapter veranschaulicht. Die Basisadresse des Erweiterungs-ROMs 40 ist im Kopfbereich 36 des Konfigurationsraumes 34 bei der Adresse 30_{16} gespeichert. Das Erweiterungs-ROM 40 enthält einen Speicher variabler Länge, der in eine Mehrzahl von Bildbereichen unterteilt ist, von denen jeder einen vorbestimmten Erweiterungs-ROM-Kopfbereich 42, einen vorbestimmten PCI-Datenstrukturbereich 44 und einen geräteabhängigen Bereich 46 enthält. Der Kopfbereich 42 beginnt bei der Adresse 00_{16} und erstreckt sich über eine Länge von 24 Byte und definiert eine Mehrzahl fester Register oder Felder. Der Kopfbereich 42 enthält einen Zeiger auf den PCI-Datenstrukturbereich 44, welcher irgendwo in den ersten 64K Bytes des ROM lokalisiert ist. Der PCI-Datenstrukturbereich 44 erstreckt sich über eine Länge von 24 Byte und bestimmt eine Mehrzahl fester Register oder Felder. Der gewählte abhängige Bereich 46 beginnt unmittelbar nach dem PCI-Datenstrukturbereich 44, erstreckt sich bis zum Ende des Bildbereiches und speichert gerätespezifische Informationen. Weitere Informationen über das Format des Erweiterungs-ROMs 40 finden sich im Dokument PCI Local Bus Specification Production Version, Reversion 2.0 vom 30. April 1993.

Wenn das Code-Typenfeld der PCI-Datenstruktur 44 einen bestimmten Wert enthält, dann enthält der geräteabhängige Bereich 46 eine PCF-Datei für den peripheren Adapter. Die Verwendung des Erweiterungs-ROMs 40 zur Speicherung der PCF-Datei erübrigt die Notwendigkeit für Disketten im Konfigurationsprozess.

Figur 4 zeigt eine Zeitdarstellung zur Veranschaulichung einer Konfigurations-Lese-Transaktion zwischen dem Prozessor und dem peripheren Adapter, wobei der Prozessor Zugriff auf Daten vom Konfigurationsraum und dem Erweiterungs-ROM des peripheren Adapters haben und diese lesen kann. Ein Konfigurations-schreibbefehl würde eine ähnliche Sequenz zum Beschreiben des Konfigurationsraumes verwenden. Für weitere Informationen über PCI-Bus-Transaktionen sei auf das Dokument PCI Local Bus Specification Production Version, Reversion 2.0 vom 30. April 1993 verwiesen, auf das hiermit Bezug genommen werde.

Ein ausgewählter Untersatz von Signalen, wie sie in Figur 4 verwendet werden, seien nachfolgend beschrieben:

CLK Das CLK-Signal (Taktsignal) sorgt für die zeitliche Abstimmung aller Transaktionen auf dem PCI-Bus und ist ein Eingangssignal für jedes Gerät. Alle anderen PCI-Bussignale werden bei der Anstiegsflanke des Taktsignales abgetastet und alle anderen Zeitsteuerparameter werden mit Bezug auf diese Flanke definiert. Es ist zu erwarten, dass der PCI-Bus über einen weiten Frequenzbereich des Taktsignales arbeiten kann.

Rahmen# Das Rahmen#-Signal (Zyklusrahmensignal) wird vom momentanen Hauptgerät vorgegeben, um den Beginn und die Dauer eines Zugriffes anzuzeigen. Das Rahmen#-Signal soll den Beginn einer Bustransaktion anzeigen. Während das Rahmen#-Signal vorhanden ist, dauern Datentransfers fort. Wenn das Rahmen#-Signal verschwindet (deassert), befindet sich die Transaktion in der Enddatenphase.

AD AD[31:00]-Signale (Adressen- und Datensignale) werden auf denselben Anschlüssen des PCI-Busses multi-

plex. Während des ersten Taktes einer Transaktion enthalten die AD-Signale eine Ziel-Geräteadresse von 32 Bit. Während der nachfolgenden Takte enthalten die AD-Signale bis zu 4 Bytes von Daten.

IDSEL Das IDSEL-Signal (Initialisierungsgerätewahl-Signal) wird während der Konfigurationslese- und -schreibtransaktionen benutzt als eine Chipwahl am Ort der oberen 24 Adressenzeilen,

C/BE# Die C/BE#[3::0]-Signale (Busbefehls- und Byteaktivierungssignal) werden an denselben Anschlüssen des PCI-Busses multiplext. Während der Adressenphase einer Transaktion definieren die C/BE#-Signale einen Busbefehl. Während der Datenphase der Transaktion werden die C/BE#-Signale als Byteaktivierungen für die 32 AD-Signale benutzt. Die Byteaktivierungen bestimmen, welche Byte-Lanes, also 8-Bitgruppen, der AD-Signale sinnvolle Daten transportieren. Beispielsweise gehört das C/BE#[0]-Signal zum Byte 0 der AD-Signale und das C/BE#[3]-Signal gehört zum Byte 3 der AD-Signale.

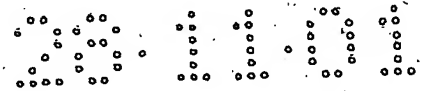
IRDY# Das IRDY#-(Initiator-Fertig)Signal zeigt die Fähigkeit des Initialisierungsgerätes (Hauptgerät) die Datenstromphase der Transaktion zu vollenden. Das IRDY#-Signal wird in Verbindung mit dem TRDY#-Signal verwendet. Eine Datenphase ist bei jedem Takt vollständig, bei welchem sowohl der IRDY#- als auch das TRDY#-Signal vorhanden (asserted) sind. Während eines Schreibvorganges zeigt das IRDY#-Signal an, dass die AD-Signale gültige Daten enthalten. Während eines Lesevorganges zeigt das IRDY#-Signal an, dass das Hauptgerät für die Datenaufnahme bereit ist.

Wartezyklen werden eingefügt, bis beide IRDY#- und TRDY#-Signale gleichzeitig vorhanden sind.

TRDY# Die TRDY#-(Zielgerät-Fertig)Signale geben an, dass das Zielgerät in der Lage ist, die gegenwärtige Datenphase der Transaktion zu vollenden. Das TRDY#-Signal wird in Verbindung mit dem oben beschriebenen IRDY#-Signal benutzt. Eine Datenphase ist bei jedem Takt beendet, bei dem sowohl das TRDY#- als auch das IRDY#-Signal vorhanden sind. Während eines Lesevorganges zeigt das TRDY#-Signal an, dass die AD-Signale gültige Daten enthalten. Während eines Schreibvorgangs zeigt das TRDY#-Signal an, dass das Zielgerät bereit zur Annahme von Daten ist. Wartezyklen werden eingefügt, bis beide IRDY#- und TRDY#-Signale zusammen vorhanden sind.

Weitere Informationen über andere Signale, die auf dem PCI-Bus benutzt werden, finden sich im Dokument PCI Local Bus Specification Production Version, Reversion 2.0 vom 30. April 1993.

In Figur 4 wird der Bus-"Besitz" einige Zeit vor der Taktperiode #1 einem Hauptgerät, beispielsweise dem Prozessor, zugeteilt. Nach dieser Besitzzuteilung wird das Rahmen#-Signal während der Taktperiode #2 erzeugt, indem es vom Prozessor auf eine niedrige Spannung gebracht wird, um den Beginn der Konfigurations-Lese-Transaktion zwischen dem Prozessor und dem peripheren Adapter anzuzeigen. Die erste Taktflanke, bei welcher das Rahmen#-Signal erscheint, also die Taktperiode #3, ist die Adressenphase, und bei dieser Taktflanke werden die Adressen- und Busbefehlscodes durch die AD- und C/BE#-Signale übertragen. Normalerweise dekodiert jedes Gerät seine eigene Adresse für normale Zugriffe. Jedoch erfordern die Zugriffe in dem Konfigurationsraum und dem Erweiterungs-ROM, dass die Geräteauswahlcodierung extern erfolgt und an den peripheren



Adapter mit Hilfe des IDSEL-Signals, welches wie ein klassisches "Chipauswahl"-Signal wirkt, signalisiert wird. Nach der Adressenphase, in welcher der Konfigurations-Lesebefehl dekodiert wird und IDSEL auftritt, also die Taktperiode #3, reagiert der periphere Adapter mit einem (nicht veranschaulichten) DEVSEL#-Signal.

Die interne Adressierung des Konfigurationsraumes und Erweiterungs-ROMs erfolgt durch die AD- und C/BE#-Signale. Die Konfigurations-Lese-Befehle (oder Schreibbefehle) erlauben Zugriffe auf ein Byte, ein Wort, eine Doppelwortgrenze, und/oder als eine Burst-Operation. Der Rest der Transaktion entspricht den anderen Befehlen, die in dem Dokument PCI Local Bus Specification Production Version, Reversion 2.0 vom 30. April 1993 beschrieben sind.

Die nächste Taktflanke, also die Taktperiode #4, lässt die erste von einer oder mehrerer Datenphasen beginnen, während welcher Daten von den AD-Signalen zwischen dem Prozessor und dem peripheren Adapter bei jeder Taktflanke übertragen werden, für welche sowohl die IRDY#- als auch die TRDY#-Signale vom Prozessor bzw. peripheren Adapter assertiert werden. In die Datenphasen können entweder vom Prozessor oder dem peripheren Adapter Wartezyklen durch Deassertieren der IRDY#- bzw. TRDY#-Signale eingefügt werden. Zu Zeiten, wenn der Prozessor nur einen weiteren Datentransfer vollenden möchte (was unmittelbar nach der Adressenphase der Fall sein kann), wird das Rahmen#-Signal deassertiert und das IRDY#-Signal assertiert, um anzuzeigen, dass der Prozessor fertig ist. Nachdem der periphere Adapter den letzten Datentransfer durch Assertieren des TRDY#-Signals anzeigt, kehrt der PCI-Bus in seinen Leerlaufzustand zurück, in welchem sowohl das Rahmen#-Signal wie auch das IRDY#-Signal deassertiert wird.

Es folgt eine Beschreibung der PCF-Syntaxspezifikation. Es sei bemerkt, dass in der Beschreibung: (1) optionale Passwörter und Felder in Klammern ([]) angegeben sind, (2) geschweifte Klammern ({}) eine Liste von Optionen angeben, worin ein senkrechter Balken (|) eine Trennung der Optionen in der Liste darstellt, und (3) Werte generell entweder in Hexadezimal- oder Binärformat angegeben werden, wobei Hexadezimalzahlen mit dem Buchstaben "h" und Binärzahlen mit "b" enden. Es versteht sich für den Fachmann, dass in der hier beschriebenen Erfindung irgendeine Anzahl unterschiedlicher Syntaxkonventionen und Passwörter benutzt werden können, und dass die nachstehend beschriebenen Syntaxspezifikation die Erfindung nicht auf die hier im einzelnen beschriebene Form beschränken soll.

LIEFERANT	xxxxh
GERÄT	xxxxh
PCFDEF	xxh

Die Lieferanten- und Gerätepasswörter müssen zu den äquivalenten Feldern im Konfigurationsraum des peripheren Adapters passen. Die PCFDEF-Parameter geben die Version der Syntaxspezifikation an.

LANG {DAN|DUT|ENG|FIN|FRE|GER|ITA|NOR|POR|SPA|SWE}

Dem Passwort LANG folgt ein Parameter, der die Sprachwahl zur Anzeige für den Operator oder Benutzer angibt, wobei DAN dänisch, DUT holländisch, ENG englisch, FIN finnisch, FRE französisch, GER deutsch, ITA italienisch, NOR norwegisch, POR portugiesisch, SPA spanisch und SWE schwedisch bedeutet. Das LANG-Passwort gibt an, dass jeder in Anführungsstriche gesetzte Text innerhalb eines folgenden Blocks die spezifizierte Sprache ist. Ein Sprachblock setzt sich fort, bis das nächste LANG-Passwort auftritt oder das Ende der PCF-Datei erreicht ist.

UDFn "User Definable Feature description" (vom Benutzer definierbare Merkmalsbeschreibung) s:aah [,s:bbh,..., s:zzh]

Der UDFn-Parameter bezeichnet ein vom Benutzer definierbares Merkmal. Das *n* wird benutzt, um ein Etikett (Label) für das UDF-Statement zu bilden. Es muss ein Wert von 0 bis 99 (dezimal) sein, und jeder Wert von *n* innerhalb eines Sprachblockes muss einmalig sein. Auf UDFn folgt die Beschreibung des Merkmals mit in Anführungsstriche gesetztem Text, dem eine Liste von Konfigurationsraumregistern folgt, die durch das Merkmal betroffen sind. Das Format der Liste enthält mindestens eine Registergrößen-/Offset-Beschreibung. Die Registergröße-/Offset-Beschreibung wird als s:xxh dargestellt, wobei s: der Größenindikator und xxh das Register/Offset vom Beginn des Konfigurationsraumes ist. Das h gibt das Offset in Hexadezimalform an. Der Größenindikator kann B: für Byte, W: für Wort oder D: für Doppelwort sein. Wenn mehr als ein Register betroffen ist, dann müssen diese nicht dieselbe Größe haben oder in irgendeiner Adressenreihenfolge vorliegen.

"Wahl 1 Beschreibung" {yyyyyyyb|zzh|zzzzh|zzzzzzzh}
[, {yyyyyyyb|zzh|...}, ...] [, OFFm, ...]

"Wahl x-1 Beschreibung" zzh[, zzzzh, ...] [, OFFm, ...]

"Wahl x Beschreibung" zzh[, zzzzh, ...] [, OFFm, ...]

Der in Anführungsstriche gesetzte Wahlbeschreibung folgt die Registerinformation. Ein Registerwert von Bytegröße kann im Binär- oder Hexadezimalformat angegeben werden. Register von Wortgröße und Doppelwortgröße können nur im Hexadezimalformat angegeben werden. Im Binärformat wird jedes Bit als ein X (oder x), 1 (Einstell-) oder 0 (Rückstell-) beschrieben werden, und ihm folgend ein "b", um anzugeben, dass es sich um einen Binärwert handelt. Im

Hexadezimalformat wird jede Tetrade (nibble) entweder als ein X (oder x) oder ein Wert von 0 bis F (oder 0 bis f) angegeben und ihm folgt ein "h", um den Hexadezimalcharakter zu bezeichnen. Das X zeigt an, dass dieses Bit oder diese Tetrade nicht modifiziert werden soll, wenn Daten zu dem spezifizierten Offset im Konfigurationsraum beschrieben werden. Diese Registercodierungen entsprechen der Reihenfolge und Größe, in welcher sie in der UDFn-Beschreibung definiert sind. Die Wahlbeschreibung kann optional auch andere UDFn-Definitionen ungültig machen, um mögliche benutzerdefinierbare Merkmale zu erlauben, wenn das Vorhandensein eines Benutzermerkmals von den zuvor getroffenen Auswahlmöglichkeiten abhängt. Bei Vorgabe wird ein UDFn aktiviert, wenn es nicht durch eine Auswahlbeschreibung speziell deaktiviert ist. Es sei bemerkt, dass eine Auswahlbeschreibung ihre entsprechende UDFn nicht deaktivieren kann. Es sei ferner bemerkt, dass ein UDFn durch Verwendung eines OFFm-Passwortes deaktiviert werden kann, wenn das m des OFFm zum n des UDFn passt.

HILFE "HILFETEXT"

Das HILFE-Passwort und der in Anführungsstriche gesetzte Text sind optional. Falls in der PCF-Datei vorhanden, muss es der letzten Wahl eines UDFn folgen.

ENDPCF

Das Passwort ENDPCF bezeichnet das Ende der PCF-Datei. Jegliche Information, die diesem Passwort ENDPCF folgt, wird nicht analysiert.

Das ;-Passwort gibt den Beginn eines Kommentars hindurch bis zum nächsten Wagenrücklauf an.

Im Folgenden wird eine Beispiel-PCF-Datei zur Veranschaulichung der Benutzung des Formats der Syntaxspezifikation angegeben:

```
LIEFERANT 1234h      ;der Lieferant-ID ist 1234h
GERÄT 5678h         ;das Gerät ID ist 5678h
PCFDEF 01h          ;PCF-Version 1
```

```
LANG ENG            ;Es folgt ein englischsprachiger Block.
```

```
UDFO "Type of Widget Communications" (Typ der Widget Kommuni-
kationen) B:45h, D:8Ch
```

```
; setzt Werte auf 45h (Byte) und 8Ch (Doppelwort)
; wenn "RF" vom Benutzer gewählt wird, dann deaktivieren
; anweisen für UDF1 und UDF2
```

```
"seriell"          xxx000xxb, ABDCXXXXh
```

```
"parallel"         xxx010xxb, 1234XXXXh
```

```
"RF"               xxx100xxb, 5678abcdh, OFF1, OFF2
```

```
HILFE "diese Wahl lässt sie auswählen, welchen Typ vom Kommu-
nikationsmechanismus sie für dieses Gerät wünschen"
```

```
UDF1 "Kommunikationsgeschwindigkeit" W:56h
```

```
"4 Mbit/Sec"       X12Xh, OFF2      ; läuft bei 4 Megabit/Sekunde
```

```
"16 Mbit/Sec"      X4cXh, OFF2      ; läuft bei 16 Megabit/Sekunde
```

```
"64 Gbit/Sec"      X00Xh            ; läuft bei 64 Gigabit/Sekunde
```

```
UDF2 "Enable Super Hyper Turbo Mode" (Aktiviert Super-Hyper-
Turbo-Modus) B:49h; setzt Byte auf 49h
```

```
"NEIN XXXXX0XXb
```

```
"JA"              XXXXX1XXb
```

HILFE "Aktiviert Super-Hyper-Turbo-Modus nur, wenn die Geschwindigkeit von 64 Gbit gewählt wurde"

UDF3 "Widget Host ID" B:9Ah

"7" xxxxx000b

"6" xxxxx001b

"5" xxxxx010b

"4" xxxxx011b

ENDPCF ; Ende der PCF-Datei.

Die Figuren 5A bis G bilden zusammen ein Flussdiagramm zur Veranschaulichung der bei der Erfindung durchgeführten Schritte.

Das Flussdiagramm der Figur 5A zeigt die durchgeführten Schritte bei der Bestimmung, ob der periphere Adapter die vom Benutzer wählbaren Konfigurationsoptionen erlaubt, und bei der Bestimmung, ob die PCF-Datei in einem Erweiterungs-ROM 40 des Adapters vorhanden ist. Der Block 46 veranschaulicht den Zugriff des Prozessors auf das PCF_erforderliche Bit in einem reservierten Byte des Kopfbereiches 36 im Konfigurationsraum 34. Der Entscheidungsblock 48 zeigt den Prozessor bei der Bestimmung, dass das PCF_erforderliche Bit im Kopfbereich 36 des Konfigurationsraumes 34 gesetzt ist. Ist dies nicht der Fall, dann endet der Konfigurationsprozess für gerätespezifische Möglichkeiten (ITEMS). Andernfalls stellt der Block 50 den Prozessor beim Zugriff auf das Erweiterungs-ROM-Basisadressenfeld im Kopfteil 36 des Konfigurationsraumes 34 dar. Der Entscheidungsblock 52 stellt den Prozessor bei der Bestimmung dar, ob der Wert des Erweiterungs-ROM-Basisadressenfeldes gültig ist. Ist dies nicht der Fall, dann geht die Steuerung auf "A" über. Anderenfalls stellt der Block 54 den Prozessor beim Zugriff auf das Codetypenfeld in der PCI-Datenstruktur 44 des ersten Bildbereiches des Erweiterungs-ROMs 40 dar. Der Ent-

scheidungsblock 56 veranschaulicht den Prozessor bei der Bestimmung, ob das Codetypenfeld einen "PCF"-Wert enthält. Wenn nicht, geht die Steuerung zum Block 48 über. Block 48 ist ein Entscheidungsblock, der bestimmt, ob zusätzliche Bildbereiche im Erweiterungs-ROM 40 vorhanden sind. Ist dies nicht der Fall, dann geht die Steuerung auf "A" über. Andernfalls geht die Steuerung zum Block 54 über, um auf das Codetypenfeld in der PCI-Datenstruktur 44 des nächsten Bildbereiches des Erweiterungsrom 40 zuzugreifen. Bestimmt der Prozessor bei Block 56, dass das Codetypenfeld einen "PCF"-Wert enthält, dann stellt der Block 60 den Prozessor beim Herausholen der PCF-Datei aus dem geräteabhängigen Bereich 46 des Erweiterungs-ROMs 40 dar. Die Steuerung geht dann auf "B" über.

Das Flussdiagramm der Figur 5B veranschaulicht die bei "A" ausgeführten Schritte, wenn die PCF-Datei sich im nicht-flüchtigen Speicher, also auf einer Festplatte oder Diskette und nicht im Erweiterungs-ROM 40 des peripheren Adapters befindet. Der Entscheidungsblock 62 stellt den Prozessor bei der Suche nach der PCF-Datei im nicht-flüchtigen Speicher dar. Vorzugsweise sollte der Dateiname der PCF-Datei XXXXYYYY.PCF sein, wobei die XXXX ein Zweibyte-Gerät Id ist, wie es im Kopfbereich 36 des Konfigurationsraumes 34 spezifiziert ist (Darstellung in hexadezimalen Zahlen), und die YYYY sind ein Zweibyte-Lieferant Id, wie er im Kopfbereich 36 des Konfigurationsraumes 34 (Darstellung in hexadezimalen Zahlen) spezifiziert ist. Findet der Prozessor die PCF-Datei im nicht-flüchtigen Speicher nicht, dann geht die Steuerung auf den Block 64 über, um den Benutzer für die Lokalisierung der PCF-Datei anzuleiten. Andernfalls geht die Steuerung über auf "B".

Das Flussdiagramm der Figur 5C zeigt die bei "B" durchgeführten Schritte, wenn die PCF-Datei entweder im Erweiterungs-ROM oder einem anderen nicht-flüchtigen Speicher gefunden wird. Der Block 66 zeigt den Prozessor beim Lesen der Lieferanten-ID.

und der Geräte-ID aus der PCF-Datei. Block 68 zeigt den Prozessor beim Speichern der Lieferanten-ID und der Geräte-ID im nicht-flüchtigen Speicher. Block 70 zeigt den Prozessor beim Lesen des PCFDEF-Passwortes und der Parameter aus der PCF-Datei. Der Entscheidungsblock 72 zeigt den Prozessor bei der Bestimmung, ob die PCFDEF sich auf einen erkannten, also gültigen Versionsidentifizierer bezieht. Wenn nicht, dann wird der Konfigurationsprozess abgebrochen. Andernfalls zeigt der Block 74 den Prozessor beim Absuchen der PCF-Datei nach LANG-Passwörtern. Der Entscheidungsblock 76 zeigt den Prozessor bei der Bestimmung, ob sich in der PCF-Datei mehr als ein LANG-Passwort befindet. Ist dies nicht der Fall, dann geht die Steuerung auf "C" über. Andernfalls zeigt der Block 78 den Prozessor bei der Anweisung an den Benutzer nach einer Sprachauswahl. Bei Block 80 akzeptiert der Prozessor die Sprachauswahl des Benutzers. Block 82 zeigt den Prozessor beim Positionieren zum Sprachblock, der durch das betreffende LANG-Passwort in der PCF-Datei identifiziert ist, zur Verarbeitung des Sprachblockes. Die Steuerung geht dann über auf "C".

Das Flussdiagramm der Figur 5D veranschaulicht die bei "C" durchgeführten Schritte, wobei die PCF-Passwörter verarbeitet und bearbeitet werden. Block 84 stellt den Prozessor beim Lesen des nächsten Passwortes und der zugehörigen Parameter aus der PCF-Datei dar. Beim Entscheidungsblock 86 bestimmt der Prozessor, ob das nächste Passwort "UDF" ist. Wenn dies der Fall ist, geht die Steuerung zur Verarbeitung des "UDF"-Passwortes und seiner Parameter auf "D" über, und danach kehrt die Steuerung zu Block 84 zurück. Beim Entscheidungsblock 88 bestimmt der Prozessor, ob das nächste Passwort "HILFE" ist. Ist es so, dann geht die Steuerung auf "E" über, um das "HILFE"-Passwort und seine Parameter zu verarbeiten, und kehrt anschließend zu Block 84 zurück. Beim Entscheidungsblock 90 bestimmt der Prozessor, ob das nächste Passwort "LANG" ist. Wenn ja, dann geht die Steuerung über auf "F". Beim Entscheidungs-

block 92 bestimmt der Prozessor, ob das nächste Passwort "PCFEND" ist. Wenn ja, geht die Steuerung über auf "F". Beim Entscheidungsblock 94 bestimmt der Prozessor, ob das Ende der PCF-Datei erreicht ist. Wenn ja, geht die Steuerung über auf "F". Andernfalls liest der Prozessor bei Block 84 das nächste Passwort aus der PCF-Datei.

Das Flussdiagramm der Figur 5E veranschaulicht die bei "D" durchgeführten Schritte sowie die Verarbeitung des UDF-Passwortes. Diese Schritte werden übersprungen, wenn der "OFF"-Parameter infolge einer früheren Benutzerauswahl gesetzt ist. Bei Block 96 zeigt der Prozessor die UDF-Beschreibung und ihre zugehörigen Auswahlmöglichkeiten an. Bei Block 98 weist der Prozessor den Benutzer zum Treffen einer Wahl an. Bei Block 100 zeigt der Prozessor die Wahl des Benutzers an, und bei Block 102 speichert der Prozessor die Auswahlwerte im nicht-flüchtigen Speicher.

Im Flussdiagramm der Figur 5F sind die bei "E" durchgeführten Schritte gezeigt sowie die Verarbeitung des "HILFE"-Passwortes veranschaulicht. Bei Block 104 zeigt der Prozessor den Hilfetext für die UDF und die zugehörigen Auswahlmöglichkeiten an.

Das Flussdiagramm der Figur 5G zeigt die bei "F" durchgeführten Schritte und veranschaulicht die Verarbeitung der vervollständigten PCF-Datei. Bei Block 106 greift der Prozessor auf den peripheren Adapter zu, und bei Block 108 aktualisiert er die entsprechenden Register im Konfigurationsraum 34 des peripheren Adapters unter Verwendung der im nicht-flüchtigen Speicher während des Konfigurationsprozesses gespeicherten Werte. Dieselben Funktionen werden auch bei der POST-Software (Power-On Self-Test = Einschalt-Eigentest) in der Anlaufzeit des Systems und während zukünftiger Systemanläufe durchgeführt, um sicherzustellen, dass die Register im Konfigurationsraum 34 korrekt initialisiert sind.

Damit ist die Beschreibung der bevorzugten Ausführungsform der Erfindung beendet. Zusammengefasst ist ein Verfahren beschrieben, welches die Konfiguration von peripheren Adaptern besorgt. Das Verfahren untersucht einen Merker im peripheren Adapter zur Bestimmung, ob vom Benutzer wählbare Konfigurationsoptionen erforderlich sind. Ist der Merker gesetzt, dann wird eine Konfigurationsdatei ausgelesen, die in einem ROM-Speicher im peripheren Adapter enthalten sein kann. Die Konfigurationsdatei wird interpretiert zur Bestimmung, ob der periphere Adapter konfiguriert werden soll und wie die Konfigurationsoptionen dem Benutzer dargeboten werden sollen. Solche Konfigurationsoptionen können konditional sein und auf früheren Auswahlen des Benutzers beruhen. Mehrere Sprachformate sind in einer einzigen Konfigurationsdatei vorhanden und erübrigen die Notwendigkeit, separate Dateien bereitzuhalten. Die Syntaxspezifikation der Konfigurationsdatei enthält einen Versionsidentifizierer für Erweiterungsmöglichkeiten und Rückwärtskompatibilität unter verschiedenen Versionen. Die Syntaxspezifikation der Konfigurationsdatei erlaubt auch das Einschreiben von Daten in den peripheren Adapter in mehreren Datenformaten. Wenn der Benutzer erst seine Auswahl getroffen hat, dann aktualisiert das Verfahren Register in den peripheren Adapter mit den gewählten Optionswerten.

PATENTANSPRÜCHE

1. Verfahren zur Konfigurierung eines mit einem Computer (12) gekoppelten peripheren Adapters (14,16,18) mit den Schritten:

a) Bestimmen, ob der periphere Adapter (14,16,18) vom Benutzer wählbare Betriebsparameter hat;

b) Zugreifen auf eine im peripheren Adapter (14,16,18) gespeicherte Konfigurationsdatei (34), wenn der periphere Adapter vom Benutzer wählbare Betriebsparameter hat;

c) Interpretieren des Inhaltes der Konfigurationsdatei zur Identifizierung von Optionen für die vom Benutzer wählbaren Betriebsparameter des peripheren Adapters (14,16,18);

d) Darstellung der Optionen für die vom Benutzer wählbaren Operationsparameter des peripheren Adapters (14,16,18) an einen Benutzer;

e) Akzeptieren einer benutzerseitigen Wahl einer der Optionen für jeden der vom Benutzer wählbaren Betriebsparameter des peripheren Adapters (14,16,18); und

f) Einstellen der Betriebsparameter des peripheren Adapters (14,16,18) entsprechend der gewählten Option.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Konfigurationsdatei (34) in einem nicht-flüchtigen Speicher (38) gespeichert ist.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass der Bestimmungsschritt (a) den Schritt eines Zugreifens auf ein Register (36) im peripheren Adapter (14,16,18) umfasst, in welchem ein Wert gespeichert ist, der angibt, ob der periphere Adapter (14,16,18) vom Benutzer wählbare Betriebsparameter aufweist.

4. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Konfigurationsdatei (34) in einer Mehrzahl von Blocks aufgeteilt ist, die mehreren Sprachen entsprechen und deren jeder Optionen für vom Benutzer wählbare Betriebsparameter des peripheren Adapters (14,16,18) enthält, die in der entsprechenden Sprache beschrieben sind.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass der Darstellungsschritt (d) die folgenden Schritte umfasst: Auswahl einer Vorgabe-Sprachwahl und Darstellung der Optionen der vom Benutzer wählbaren Betriebsparameter des peripheren Adapters (14,16,18) an den Benutzer, von dem der Vorgabe-Sprachwahl entsprechenden Block.

6. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass der Darstellungsschritt (d) die Schritte umfasst: Anweisung des Operators zu einer Sprachwahl, Akzeptieren der Sprachwahl des Benutzers und Darstellung der Optionen der vom Benutzer wählbaren Betriebsparameter des peripheren Adapters (14,16,18) an den Benutzer, von dem der Sprachwahl entsprechenden Block.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Konfigurationsdatei (34) einen Versionsindikator enthält.

8. Verfahren nach Anspruch 7, gekennzeichnet durch den Schritt der Interpretierung des Inhaltes der Konfigurationsdatei entsprechend dem Versionsindikator.

9. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die Konfigurationsdatei (34) ein oder mehrere konditionale Optionen enthält, und dass der Darstellungsschritt (d) den Schritt der Darstellung der Optionen der vom Benutzer wählbaren Betriebsparameter des peripheren Adap-

ters (14,16,18) an den Benutzer aufgrund einer früher akzeptierten Benutzerwahl entsprechend der Konditionsoption umfasst.

FIG. 1

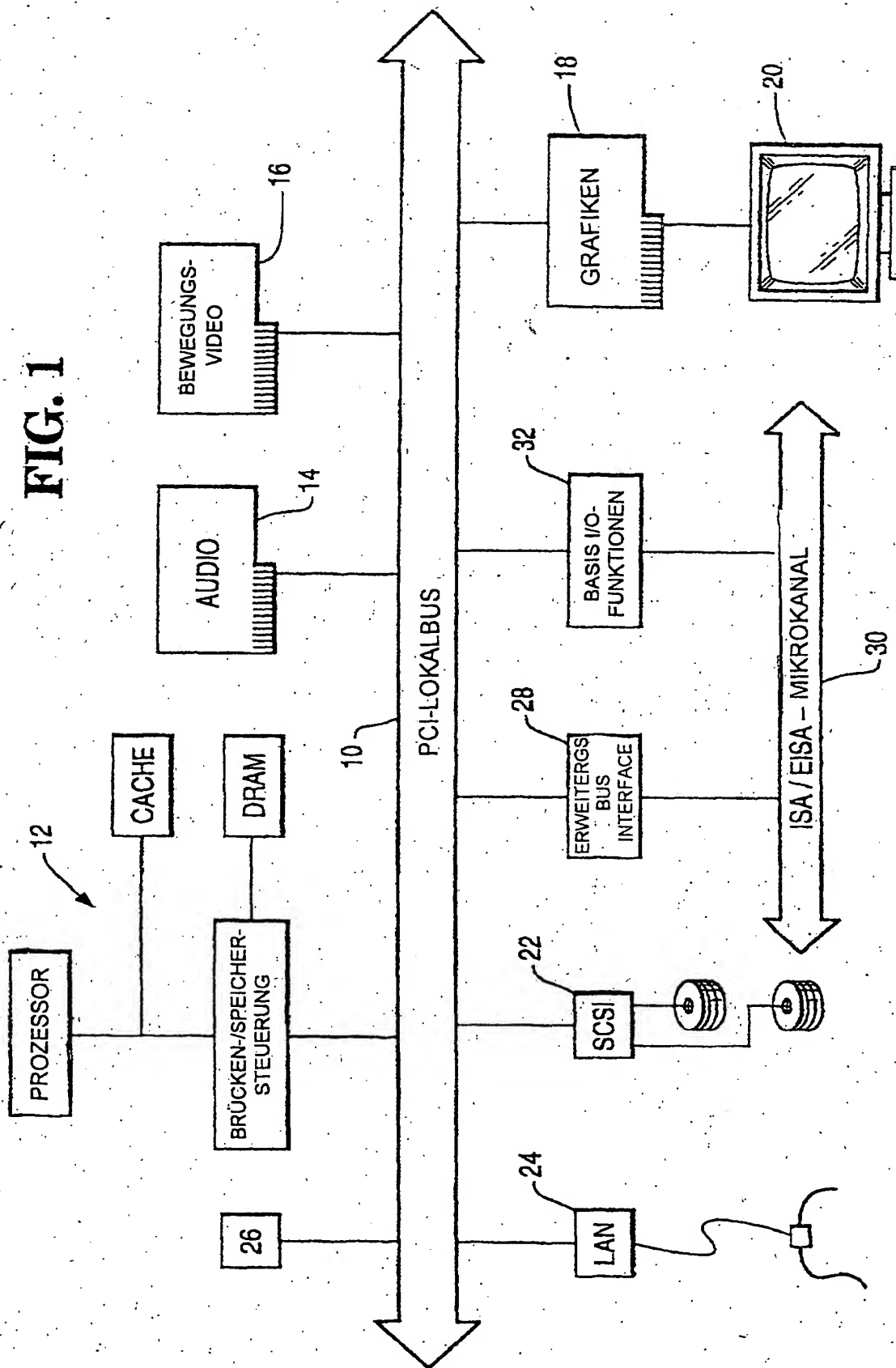


FIG. 2

2/9

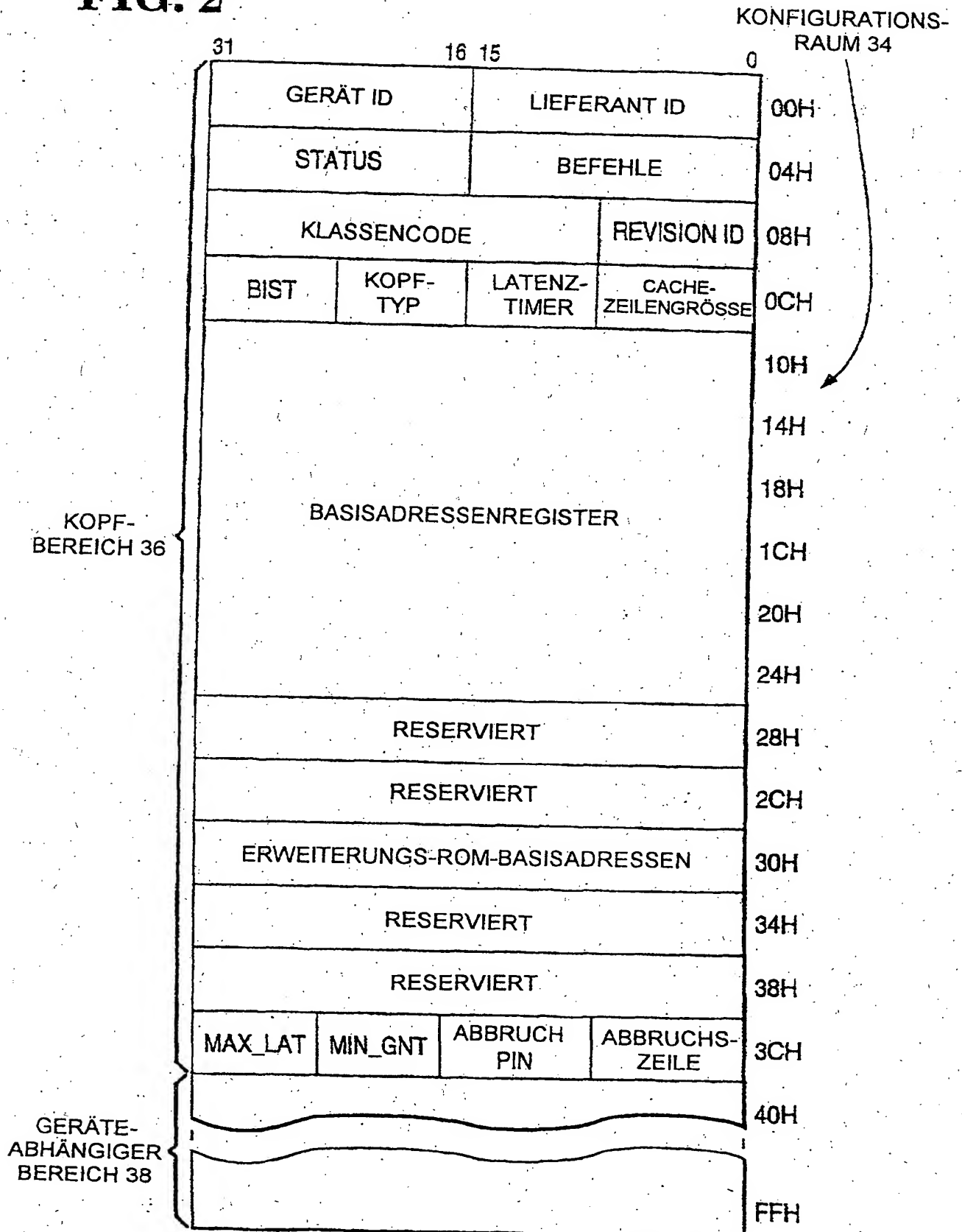
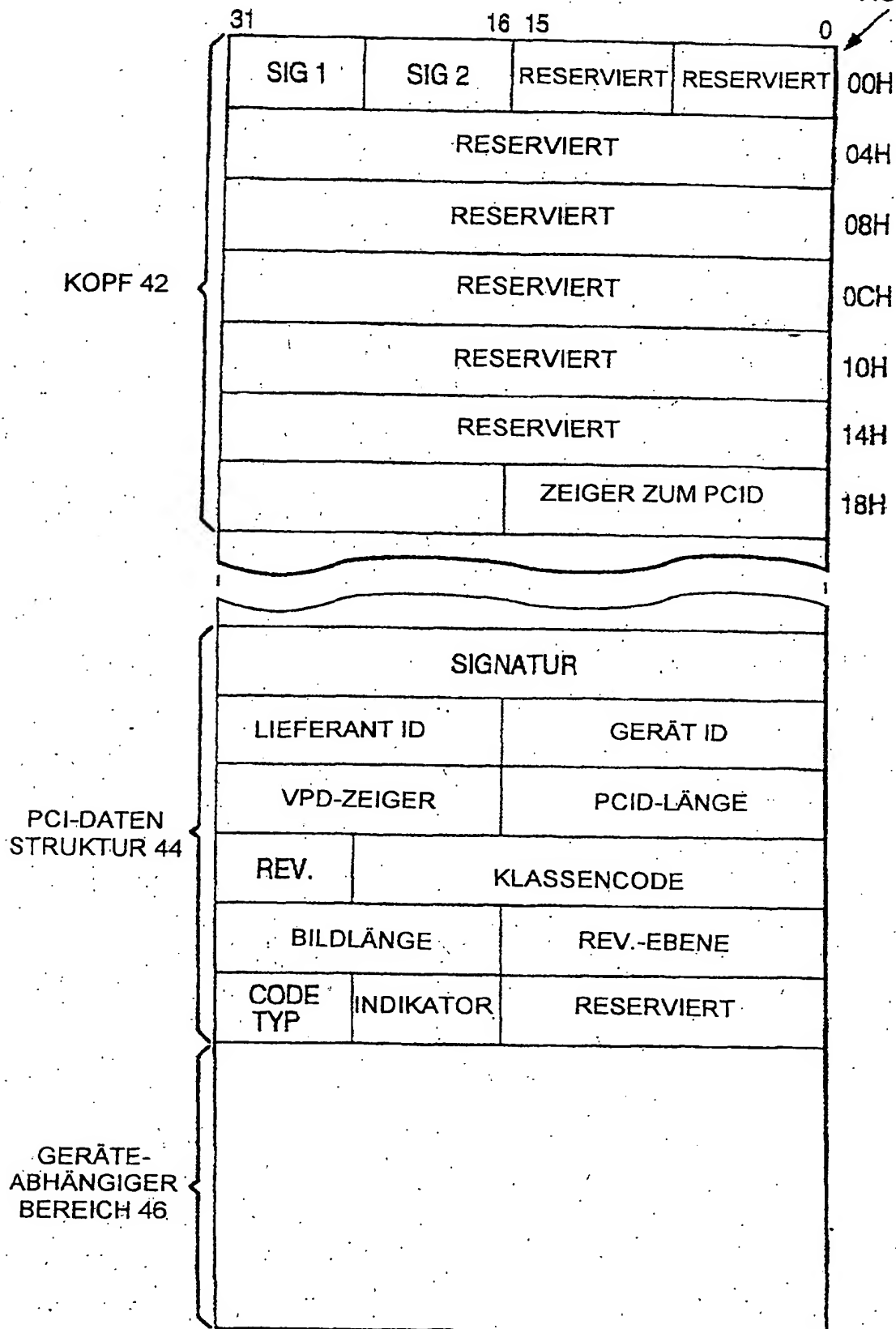


FIG 3

3/9

ERWEITERUNGS-
ROM 40

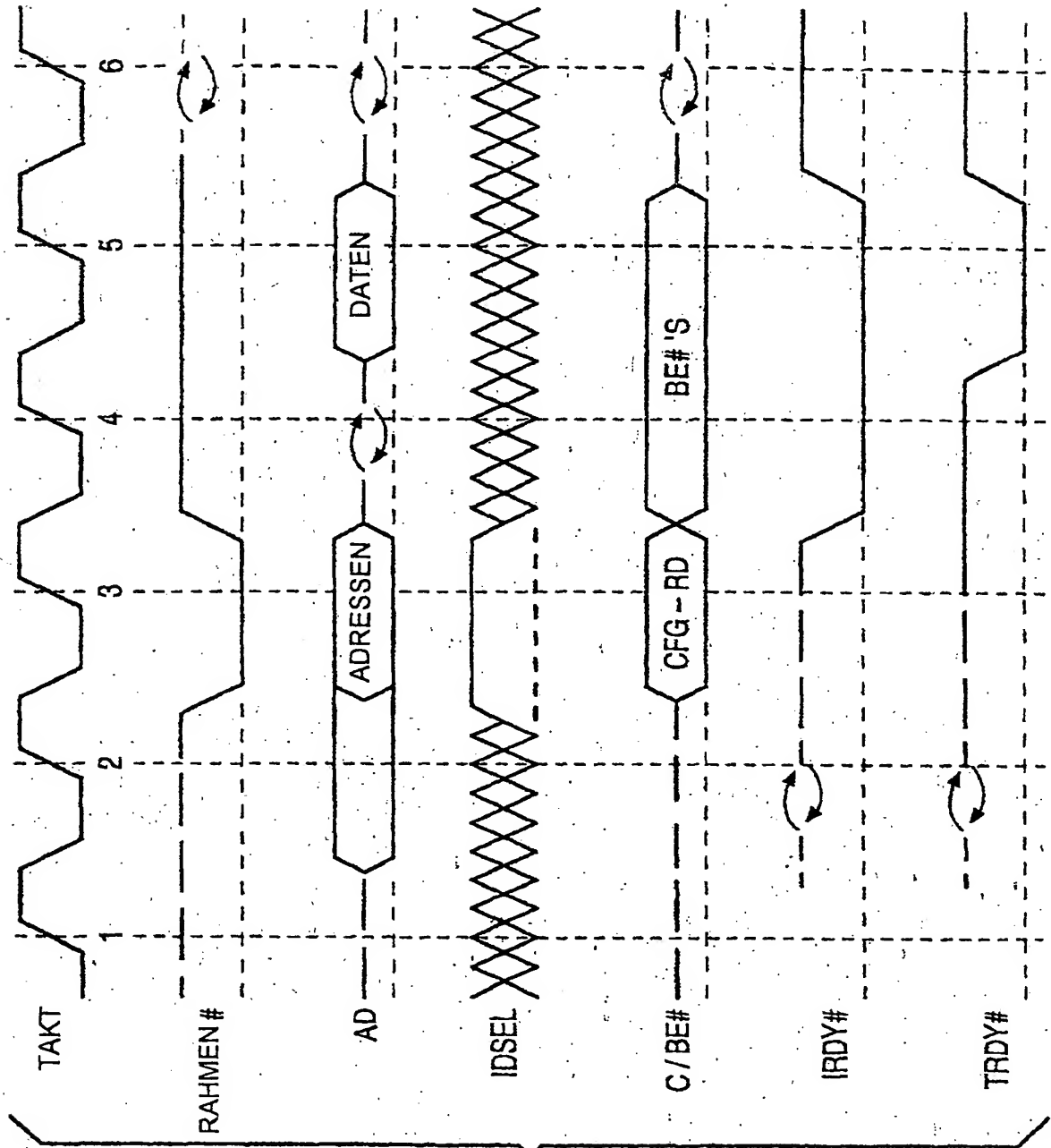


FIG. 4

FIG. 5A

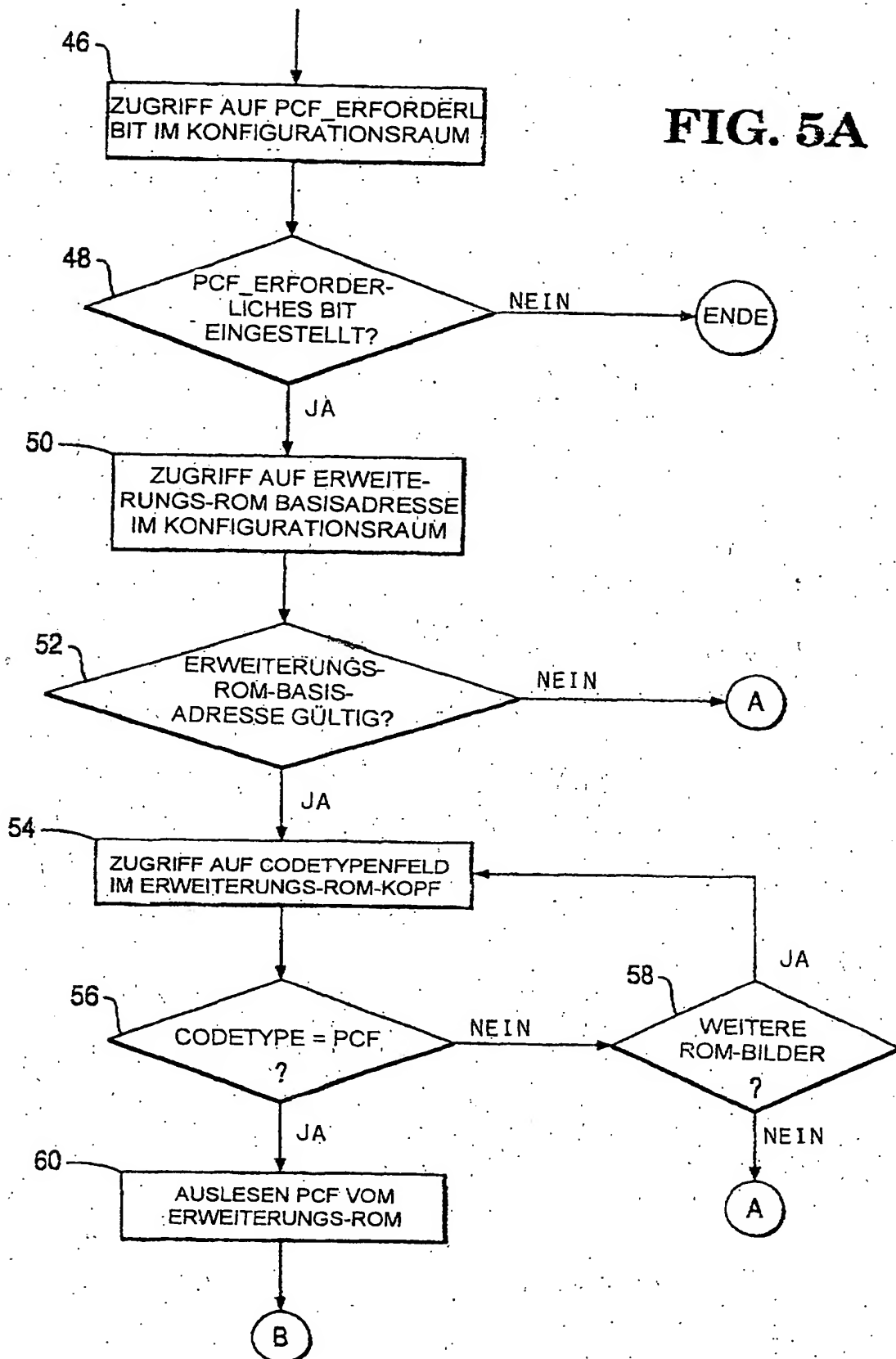
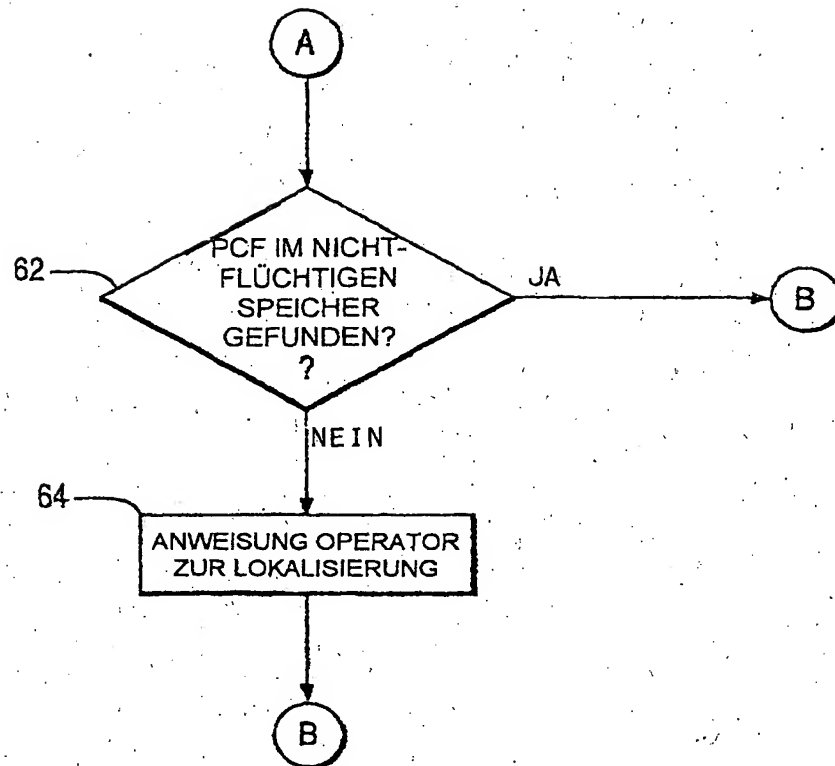


FIG. 5B



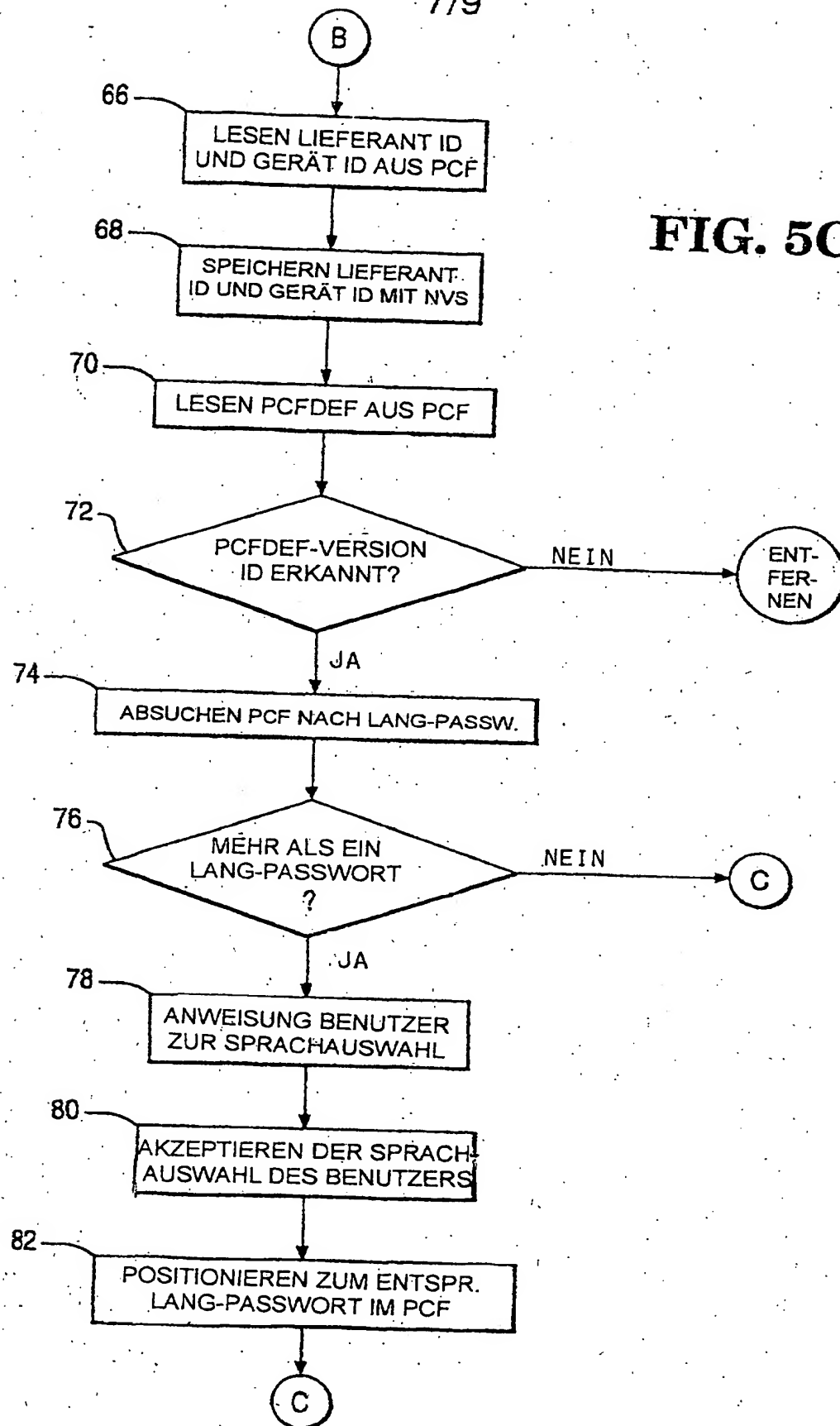


FIG. 5D

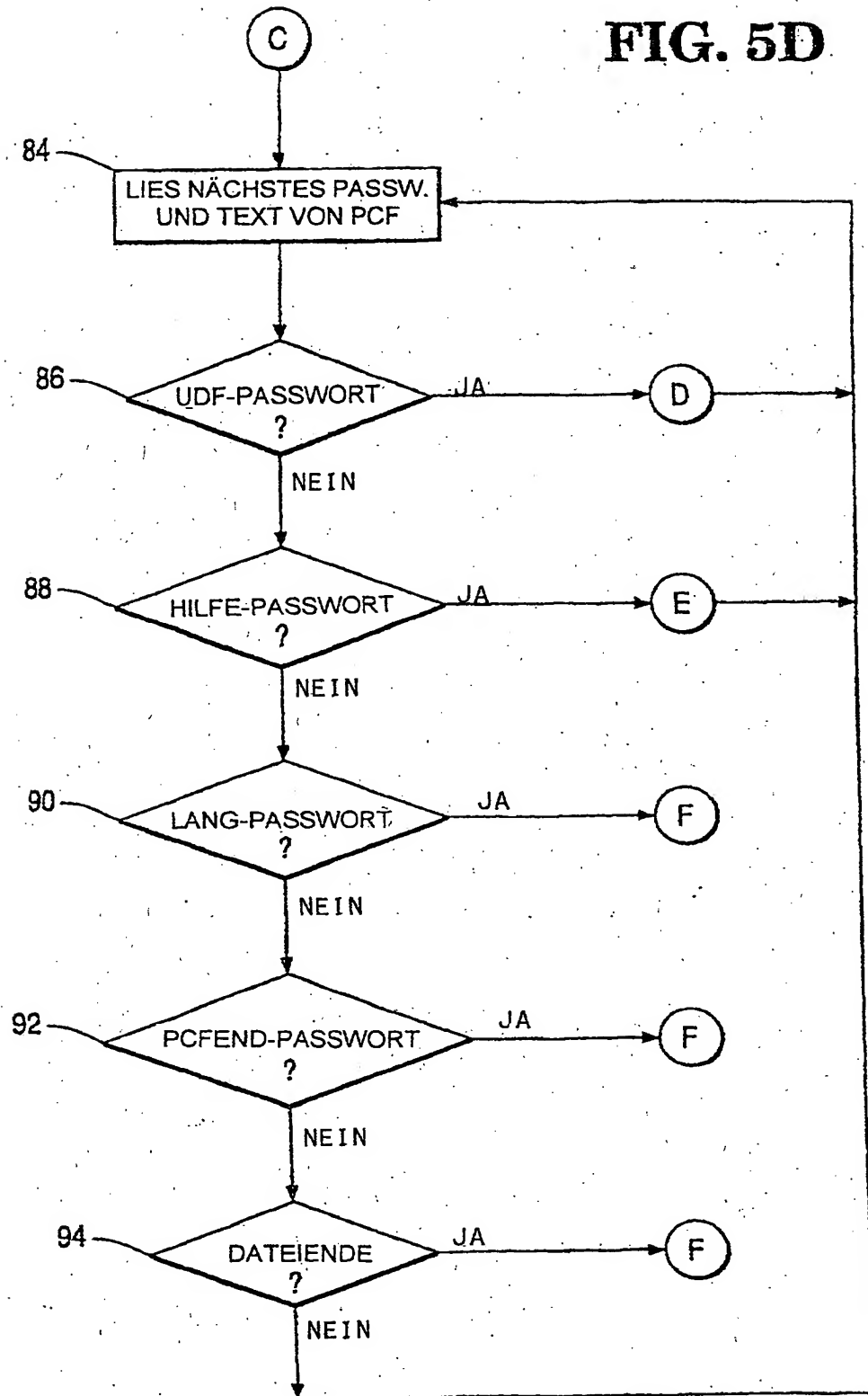
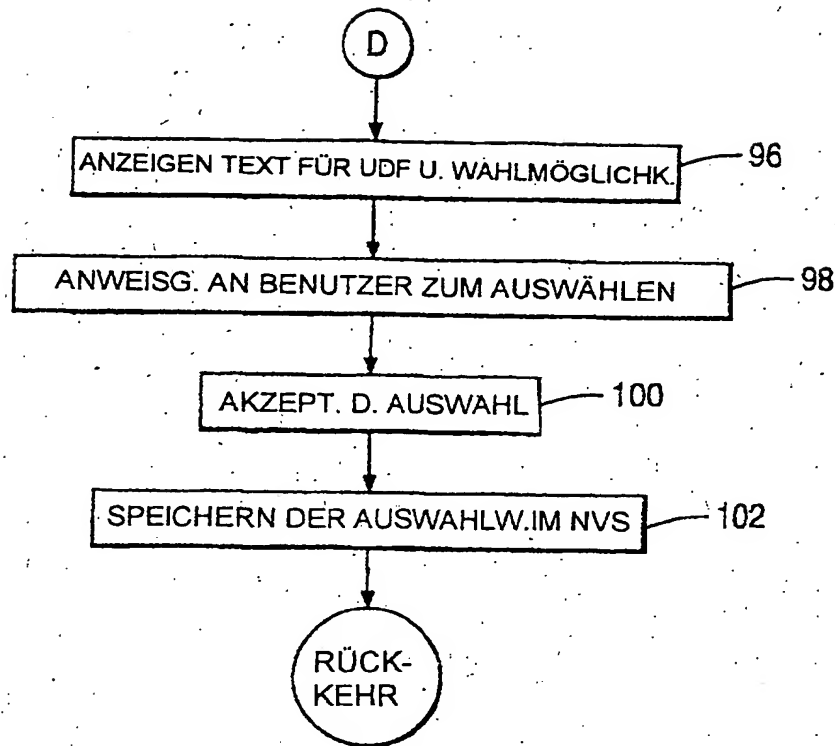
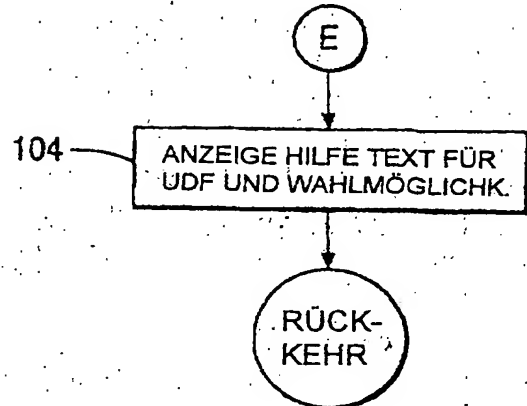


FIG. 5E**FIG. 5F****FIG. 5G**